side			result set
DB=USPT; PLUR=YES; OP=ADJ			
L3	(volatile with (non-volatile or nonvolatile) with (processor or cpu)) same (generat$ with (identifier$ or ID))	20	L3
L2	(volatile with (non-volatile or nonvolatile) with (processor or cpu)) and (generat$ with (identifier$ or ID))	533	L2
L1	(volatile same (non-volatile or nonvolatile) same (processor or cpu)) and (generat$ with (identifier$ or ID))	872	L1

END OF SEARCH HISTORY

Refine Search

Search Results -

Term	Documents
VOLATILE	164843
VOLATILES	21840
NON-VOLATILE	59301
NON-VOLATILES	1061
NONVOLATILE	32912
NONVOLATILES	615
PROCESSOR	267135
PROCESSORS	83619
CPU	132769
CPUS	10519
ID	142195
((VOLATILE WITH (NON-VOLATILE OR NONVOLATILE) WITH (PROCESSOR OR CPU)) SAME (GENERAT$ WITH (IDENTIFIER$ OR ID))).USPT.	20

There are more results than shown above. Click here to view the entire set.

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L3

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Friday, May 13, 2005 Printable Copy Create Case

Set Name side by	Query	Hit Count	Set Name

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L3: Entry 2 of 20 File: USPT Jun 15, 2004

DOCUMENT-IDENTIFIER: US 6751667 B1
TITLE: System for generating unique identifiers in a computer network

Detailed Description Text (2):
FIG. 1 is a block diagram showing the context of the volatile and non-volatile memory components of the present system 100 in an operational context. As shown in FIG. 1, non-volatile memory element 101 and volatile memory element 102 are coupled to a processor 103 which utilizes the two memory elements to generate identifiers which are unique across time and space. These identifiers are hereinafter referred to as Universally Unique Identifiers, or `UUIDs`. In a large network it is typically necessary to unambiguously differentiate between large numbers of objects, thus necessitating a very large number space. The present system provides a mechanism for generating an extremely large range of numbers (on the order of 2.sup.48, in one embodiment of the invention) while requiring the associated non-volatile memory to undergo a relatively small number of erase cycles.

CLAIMS:

1. A method for creating a series of unique identifiers using a processor coupled to volatile memory and to at least one block of non-volatile memory, the method comprising the steps of: storing a predetermined identifier in said non-volatile memory and in said volatile memory; setting a bit string in said non-volatile memory to a value of all ones; setting a number subfield and a range subfield, together comprising an extension field in said volatile memory, to zero; and generating a monotonic sequence of said unique identifiers by repetitively performing the steps of: incrementing said number subfield; creating said unique identifier by concatenating said predetermined identifier and said extension field; and when said number subfield contains all ones, performing the steps of: setting to zero, a next sequential bit in the bit string in said non-volatile memory; incrementing said range subfield in said volatile memory; and resetting said number subfield to zero.

4. A method for creating a series of unique identifiers using a processor coupled to volatile memory and to at least one block of non-volatile memory, the method comprising the steps of: storing a predetermined identifier in said non-volatile memory and in said volatile memory; setting a bit string in said non-volatile memory to a value of all ones; setting a counter in said non-volatile memory to a value of zero; setting a number subfield and a range subfield, together comprising an extension field in said volatile memory, to zero; and generating a monotonic sequence of said unique identifiers by repetitively performing the steps of: incrementing said number subfield; creating said unique identifier by concatenating said predetermined identifier and said extension field; and when said number subfield contains all ones, performing the steps of: incrementing said range subfield in said volatile memory; and resetting said number subfield to zero; setting to zero, a next sequential bit in the bit string in said non-volatile memory; and when said bit string in said non-volatile memory contains all zeroes, performing the steps of: incrementing counter in said non-volatile memory; and resetting said bit string in to all ones.

7. A system for generating a series of unique identifiers for use in a computer network, the system comprising: volatile memory containing one of the unique identifiers comprising a predetermined identifier and a field including a number subfield and a range subfield; non-volatile memory containing a copy of said predetermined identifier and bit string representing a value of said range subfield; and a processor coupled to said volatile memory and said non-volatile memory; wherein said system generates a monotonic sequence of said unique identifiers by incrementing said number subfield.

11. A system for generating a series of unique identifiers for use in a computer network, the system comprising: volatile memory containing one of the unique identifiers comprising a predetermined identifier and a field including a number subfield and a range subfield; non-volatile memory containing a copy of said predetermined identifier and bit string representing a value of said range subfield; and a processor coupled to said volatile memory and said non-volatile memory; wherein said system generates a monotonic sequence of said unique identifiers by incrementing said number subfield; wherein, when said number subfield contains all ones, a bit in the bit string in said non-volatile memory is set to zero; said range subfield is incremented; and said number subfield is reset to zero; and wherein, a value for said one of the unique identifiers is determined by storing, in said range subfield, the binary equivalent of the number of zero bits in the bit string in said non-volatile memory.

13. A system for generating a series of unique identifiers for use in a computer network, the system comprising: volatile memory containing one of the unique identifiers comprising a predetermined identifier and a field including a number subfield and a range subfield; non-volatile memory containing a counter, a copy of said predetermined identifier and bit string representing a value of said range subfield; and a processor coupled to said volatile memory and said non-volatile memory; wherein said system generates a monotonic sequence of said unique identifiers by incrementing said number subfield; wherein, when said number subfield contains all ones, a bit in the bit string in said non-volatile memory is set to zero; said range subfield is incremented; and said number subfield is reset to zero; and wherein, when said bit string in said non-volatile memory contains all zeroes, the counter in said non-volatile memory is incremented and the bit string in said non-volatile memory is set to all ones.

Previous Doc Next Doc Go to Doc#

US006751667B1

(12) **United States Patent**
Helliwell

(10) **Patent No.: US 6,751,667 B1**
(45) **Date of Patent: Jun. 15, 2004**

(54) **SYSTEM FOR GENERATING UNIQUE IDENTIFIERS IN A COMPUTER NETWORK**

(75) Inventor: **Richard P. Helliwell**, Colorado Springs, CO (US)

(73) Assignee: **Hewlett-Packard Development Company, L.P.**, Houston, TX (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 745 days.

(21) Appl. No.: **09/680,678**

(22) Filed: **Oct. 6, 2000**

(51) **Int. Cl.**[7] **G06F 1/02**
(52) **U.S. Cl.** **709/226**; 708/250
(58) **Field of Search** 709/226; 708/250

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,819,818 A	*	4/1989	Simkus et al.	463/22
5,541,996 A	*	7/1996	Ridenour	380/46
5,778,069 A	*	7/1998	Thomlinson et al.	380/262
6,243,738 B1	*	6/2001	Hayles et al.	709/203
6,324,558 B1	*	11/2001	Wilber	708/255
6,490,637 B1	*	12/2002	Shih	710/8
6,532,497 B1	*	3/2003	Cromer et al.	709/250
6,533,664 B1	*	3/2003	Crumby	463/42
6,628,786 B1	*	9/2003	Dole	380/44

* cited by examiner

Primary Examiner—David Y. Eng

(57) **ABSTRACT**

A system for generating unique identifiers (UUIDs) for software objects and other components in a network in which a large number of components may exist simultaneously and/or over a period of time. UUIDs generated by a particular product are divided into two sub-fields. One sub-field is stored in (relatively slow) non-volatile memory, and incremented infrequently. The other sub-field is stored in relatively fast, volatile RAM, that can be incremented quickly. During operation, the product creating the UUIDs generates new UUIDs by incrementing the field stored in RAM. When overflow of the RAM field occurs, the field stored in non-volatile memory is incremented. A block of flash memory is initialized to all ones, and the bits therein are then sequentially cleared to generate each subsequent unique identifier. The present system provides the equivalent of a counter that can count up to the number of available bits in non-volatile memory plus one, while reducing the number of flash memory erase cycles to one cycle for each time all the bits are cleared.

15 Claims, 5 Drawing Sheets